2014 IEEE 38th Annual International Computers, Software and Applications Conference Workshops

The Concept of a Mobility Data Cloud: Design, Implementation and Trials

Nikolay Tcholtchev, Benjamin Dittwald, Thomas Scheel, Begüm İlke Zilci, Danilo Schmidt, Philipp Lämmel, Jurma Jacobsen, and Ina Schieferdecker
Fraunhofer Institute for Open Communication Systems FOKUS, Berlin, Germany
{firstname.lastname}@fokus.fraunhofer.de

***Abstract*— The *mobility data cloud* is an Internet based platform for the integration of mobility related data from different data providers. It offers a framework for the acquisition, aggregation, provisioning, and analysis of mobility relevant data originating from various sources, e.g. charging stations, gas stations, position of vehicles, battery charge condition of electric vehicles, utilization of vehicles, and data regarding public transport. These various types of data should facilitate the collaborative shared utilization of mobility resources such as electric vehicles, charging stations, etc. This paper presents the design and implementation of such an Internet based *mobility data cloud*. In addition, the presented prototype is evaluated based on trials in a European city.**

***Keywords* — Cloud Services, Electric Mobility, Smart Cities, Web Services, Internet based Platforms**

I. INTRODUCTION & MOTIVATION

IN today's mobility concepts, the opportunities of the interconnected society and infrastructures play only a minor role so far. Such interconnections offer a lot of potential for improving the environmental, social and economic aspects of quality of life in cities. Our approach aims at establishing innovative processes and applications for the efficient shared use of different heterogeneous mobility related resources, whilst at the same time improving the quality of life with respect to mobility. In the scope of the research presented here, a cloud based framework is devised that enables the efficient interaction of all relevant technologies and platforms such as smartphones, (electric) vehicles, REST services, charging/gas stations, and navigation systems. To achieve this, both open and access restricted mobility data - currently available in a highly fragmented, heterogeneous, and unstructured way - are provided and aggregated via open interfaces, based on Internet standards for data exchange, such as JSON and XML. This requires scalable cloud-based infrastructure and services, which facilitate the access to mobility data. Based on such considerations, we present the concept of a *mobility data cloud*. The mobility data cloud is intended to provide free access and processing of mobility data and to offer services and interfaces to app developers. The goal is to create an open ecosystem of data sources, data providers, and application/service developers.

The rest of this paper is organized as follows: Section II presents the concept of a mobility data cloud and introduces the belonging architecture. Section III describes a user perspective scenario for the utilization of the mobility data cloud for electric mobility. Section IV elaborates on the prototype implementation. Section V describes the trials, which were executed in the city of Berlin, and presents the obtained numerical results. The following section VI discusses on related research efforts and implementations. Finally, section VII draws conclusions and outlines future research directions.

II. MOBILITY DATA CLOUD

A. Mobility Data Cloud Concept

The mobility data cloud is an Internet based platform for the integration of mobility related data from different data providers. It offers a framework for the acquisition, aggregation, provisioning, and analysis of mobility relevant data originating from various sources, e.g. charging stations, position and battery charge condition of electric vehicles, utilization of vehicles, and data regarding public transport.

The mobility data cloud is largely based on the concept of Open Data, which prescribes that data should be made freely available in machine readable formats. Beyond this, the mobility data cloud should be openly accessible w.r.t. to the implementation of services by developers from the community. This implies that the mobility data cloud is intended to provide a sort of a Platform-as-a-Service (PaaS) that should enable (trusted) external developers to deploy services which have been implemented on the basis of the data, provided and aggregated by the mobility data cloud. Through the opening of the mobility data cloud on the level of PaaS, a new open platform emerges for the efficient handling of mobility relevant data and deployment of services on top of it.

B. General Architecture

In the following, the different architectural aspects of the mobility data cloud are elaborated. The overall architecture is illustrated in **Figure 1**. On the left side one can see examples for the various data sources which are integrated into the data cloud. These may include Map Resources, Governmental Road Data, Vehicle Provider Data, etc. These various types of data are integrated and offered over the mobility data cloud, such

978-1-4799-3578-9/14 $31.00 © 2014 IEEE
DOI 10.1109/COMPSACW.2014.36

that applications/services can be developed on top of it. These applications are illustrated on the right side in **Figure 1** and may include different types of (mobile) apps and services utilizing the data from the cloud, for instance embedded apps in a vehicle's on-board unit, or web applications which can be accessed over a web browser.

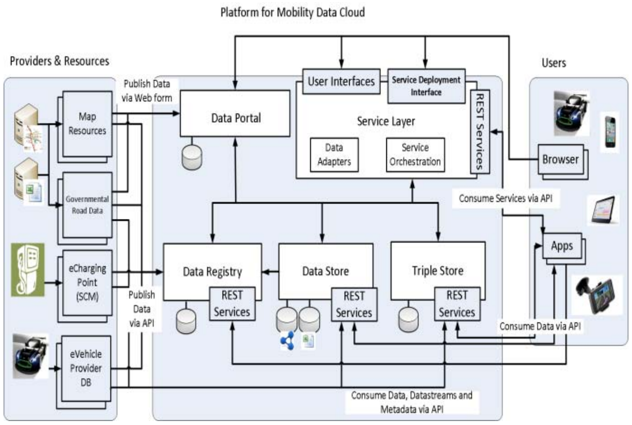

FIGURE 1. OVERALL ARCHITECTURE OF THE MOBILITY DATA CLOUD

The platform for the mobility data cloud is depicted in the center of **Figure 1**. Following components constitute the core of the platform: 1) Data Portal, 2) Data Registry, 3) Data Sore, 4) Triple Store, and 5) Service Layer. Thereby, the mobility data cloud builds on the Open Data Platform which has emerged within the Open Cities project [10]. Therefore, the mobility data cloud can be seen as an extensive add-on to this open source solution [9] .

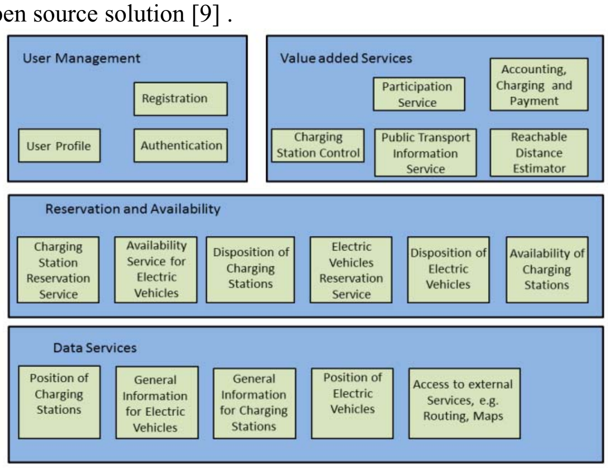

FIGURE 2. SERVICES FOR ELECTRIC MOBILITY

Next, each of the mobility data cloud's modules is described in turn:

Data Portal: The data portal is one of the components which are also present within the aforementioned Open Data Platform from the EU Open Cities project. The task of the Data Portal is to present the data of the mobility data cloud to potential users of that data, e.g. app developers or data journalists. In addition the Data Portal offers the possibility to provide new data and data sources for the mobility data cloud. In that context, it must be mentioned that the mobility data cloud can either keep/store mobility data within its architecture, or just refer to an external source for this data. In addition, the portal offers social media features for advanced (open) data users, such as comments to datasets, and possibility for forum discussions on topics around the provided mobility data (e.g. new required data sets, quality of the data accessible over the data cloud etc.).

Data Registry: The Data Registry is the component responsible for cataloguing and managing the data resources within the mobility data cloud. That is, the Data Registry keeps a number of meta-data entries that describe the data which is accessible over the mobility data cloud. This includes data which is provided by external sources as well as data which is stored or imported into the mobility data cloud.

Data Store: The data store is a component which is designed to store arbitrary data by providing a REST interfaces to an underlying data base. This component can be either realized using a NoSQL [12] data base or by providing an open REST interface to an underlying relational data base and allowing the utilizing applications and services to dynamically define the format (SQL tables and fields) for the data exchange/storage.

Triple Store: The Triple Store is another component for storing data within the mobility data cloud. It offers the possibility to store ontology based linked data within the platform and access it over a REST interface, e.g. by using the SPARQL query language over a SPARQL endpoint.

Service Layer: The Service Layer is the module that allows for deploying mobility services/applications on top of the data accessible within the mobility data cloud. This takes place over the *Service Deployment Interface* that allows for trusted developers to add new services or to update existing services utilizing multiple datasets over the mobility data cloud platform. That way the aforementioned opening of the platform on PaaS level is realized. These services may use data from external data sources which has been catalogued in the Data Registry (data providers can input such data to the Data Registry over the Data Portal). In such case, some use case specific Data Adapters would be required, in order to transform the data from its specific "raw" format into another format that can be used by the service in question. In addition, the services/applications may also offer specific user interfaces, e.g. in the case of web based mobile applications, which are also accessible over the HTTP protocol as part of the Service Layer. The services also provide REST interfaces, and can be accessed over these JSON/XML based APIs from various devices with an IP data connection (e.g. Internet), for instance smartphones using a 3G or 4G network, as well as corresponding embedded devices such as on-board units and sensors. Finally, the single services can be combined as to provide more complex services. This can be done with the help of the *Service Orchestration* module which would be capable of creating chains of services to be executed in a row and pass their input among each other. Such service orchestration components are common in the WSDL domain.

Based on the above descriptions, the next subsection presents a set of services, which were devised for the purpose of electric mobility in the course of the GeMo project [13].

C. Service Layer for Electric Mobility in Smart Cities

Figure 2 presents the Service Layer of the mobility data cloud with the services which were devised for the realization of collaborative electric mobility in smart cities. These services result from various use cases that were captured in the course of the GeMo project which deals with collaborative electric mobility and fleet management. The services are organized in a layered architecture that starts from the bottom with the *Data Services* and proceeds with the services for *Reservation and Availability* of the electric mobility resources (i.e. electric vehicles, charging stations etc.). Finally, on top of the service stack in **Figure 2**, one can observe the *User Management* services and the *Value added Services*. The User Management services handle aspects related to the clients - customers as well as applications - accessing the electric mobility services. Examples of such aspects are given by tasks such as *Registration*, *User Profile* management and *Authentication*. The Value added Services contain functionality, which is beyond the mere management of the electric mobility resources and infrastructure. For example, they would combine the data from the mobility data cloud with important information/services relating to other mobility resources, which are not necessarily tight to electric mobility, such as *Accounting, Charging and Payment*, e.g. over a special external ticketing or payment system.

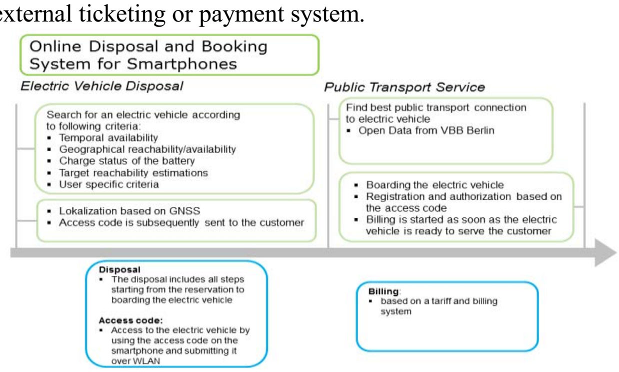

FIGURE 3. ACCESSING AND ELECTRIC VEHICLE OVER THE MOBILITY DATA CLOUD

Various interactions are required among the services in **Figure 2** in order to realize different scenarios for electric mobility. In the scope of the GeMo project, a large number of (message) sequence diagrams were captured based on an extensive use case analysis. These sequence diagrams enable the realization of key processes and activities in the scope of electric mobility for smart cities. In that line of thoughts, the following section elaborates the user perspective of a complete electric mobility scenario.

III. SCENARIO

The coming figures describe the user perspective of the mobility data cloud utilization for the purpose of electric mobility in smart cities. Figure 3 describes the steps in the scope of the disposal and reservation of an electric vehicle over a smartphone. The descriptions above the arrow elaborate the process itself, whilst the explanations underneath the arrow clarify on aspects of this process. The user experience starts with the search for an electric vehicle according to a number of criteria, such as geographical reachability, temporal availability, target reachability estimations and other user specific criteria. Once a suitable vehicle has been found and reserved by the customer, an access code is generated by the corresponding service of the mobility data cloud and sent back to the smartphone of the customer. This access code is later used during the WLAN communication with the electric vehicle in order to unlock it. Having reserved an electric vehicle, the customer/user has to reach it. This can be done by using the mobility data cloud service for public transport information (see Figure 3). In the scope of the GeMo project, this service was realized through the usage of (open) data from the Berlin public transportation provider (BVG and its umbrella association named VBB). After reaching the electric vehicle, the customer boards it and can begin her/his journey.

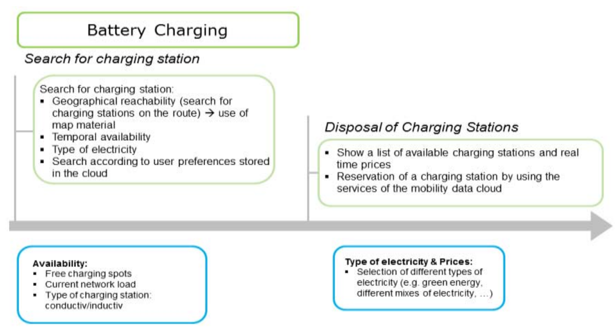

FIGURE 4. SUPPORT FOR BATTERY CHARGING OVER THE MOBILITY DATA CLOUD

Depending on the battery charge condition, it may be required that the customer uses a charging station if the vehicle is reserved for a longer period by that customer (the charging might also happen overnight). In such case, the customer would need to go through the steps described in **Figure 4**. First, a charging station needs to be searched for over the app on the customer's smartphone. As in the case of the electric vehicle disposal, various constraints would need to be considered, such as geographical reachability, temporal availability, type of electricity (e.g. percentage of green energy), or any special type of user preferences stored in the mobility data cloud. The search is conducted over the *Availability of Charging Stations* service of the mobility data cloud, and a list of charging stations matching the users' criteria is displayed on the customer's smartphone. Subsequently, a suitable charging station is reserved over the

Charging Station Reservation Service, and the charging of the vehicles battery can be triggered.

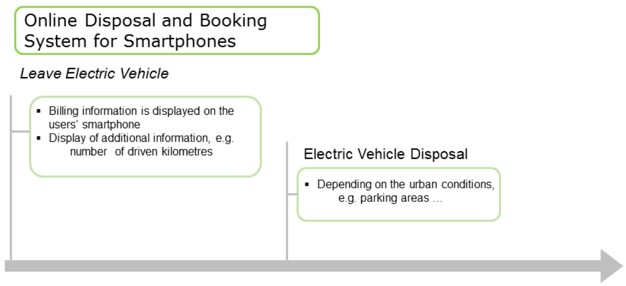

FIGURE 5. ELECTRIC VEHICLE DISPOSAL

The last step in the presented simplified utilization is constituted by the disposal of the electric vehicle. This is described in **Figure 5**, and consists of the customer leaving the electric vehicle and receiving the billing information for the time she/he has been using the vehicle. This information is displayed on the customer's smartphone and requires to be confirmed by the end user. Moreover, the disposal takes place in compliance to additional information/constraints supplied by the mobility data cloud, such as parking lots and various urban conditions (e.g. parking taxes etc.).

The scenario presented here, even though far from capturing all possible use cases and flows, exemplifies how a customer would use the mobility data cloud in the scope of collaborative electric mobility as perceived within the GeMo project [13].

IV. IMPLEMENTATION

Having presented the various architectural aspects of the mobility data cloud, next some implementation details are elaborated. The Data Portal of the mobility data cloud (see **Figure 1**) was implemented using the Liferay Portal software [15] that is easily extensible through the use of Java based portlets [16], which are implemented according to JSR168. Liferay provides various portal features such as forum, social media features, user management system etc. In addition, the Data Registry is realized based on the CKAN (Comprehensive Archive Knowledge Network) [18] solution provided by the Open Knowledge Foundation [19]. CKAN offers a widely accepted API for storing and retrieving meta-data, i.e. entries which describe the actual data(sets) in the mobility data cloud. As previously mentioned, the data itself resides either on the remote system of the data providers or is stored within one of the belonging components of the mobility data cloud, i.e. Data Store or Triple Store. The integrated version of Liferay and CKAN constitutes the Open Data Platform of the Open Cities project [9][10], which was also used as a basis for the governmental data platform for Germany [11][14].

In addition, the components for storing data within the mobility data cloud need to be elaborated. These are the Data Store and the Triple Store (see **Figure 1**). As for the Data Store, a Java based component was implemented that runs within a Tomcat Servlet Container [20]. This component provides a REST interface on the basis of the Jersey framework [17]. Thereby, the REST interface allows to upload data in formats such as CSV, XML, and JSON. The uploaded data is subsequently analyzed and automatically stored in the underlying database (currently PostgreSQL [21] is used) by using the Hibernate persistency framework [26]. Thereby, "automatically" means that suitable SQL statements are extracted based on the preceding data analysis. In addition, the data in the storage can be queried over the REST interface by using the Hibernate Query Language (HQL) [22], which is an SQL based language for querying databases through the Hibernate persistency framework. Thereby, Hibernate takes care of aspects such as transactions and synchronization of multiple parallel/concurrent data requests/updates. Furthermore, the Triple Store component is realized through the use of the Virtuoso [24] database engine that provides the functionality for storing RDF (Resource Description Framework) graphs and querying those over a REST interface by using the SPARQL query language.

The Data Store and the Triple Store are used as the core components for storing mobility relevant data in the mobility data cloud. This data is further utilized by the services running within the *Service Layer*. The various services were again implemented in Java and run as servlets within a Tomcat Server. These services use the Spring [28] and Jersey frameworks and communicate among each over REST. Thereby, JSON is widely used as the format for communication among the mobility data cloud services. In addition, the services use the HQL language for accessing the data within the Data Store.

As previously mentioned, the mobility data cloud should be opened not only on the level of data - in terms of open APIs for accessing mobility data and services, but also on the PaaS level, such that trusted developers from the community can easily add new services or update the functionality of existing services. For this purpose, trusted developers would be given access to a Git [25] repository, on top which the Jenkins [29] continuous integration platform takes care of compiling, integrating, and testing (based on predefined tests cases) the services, before deploying them to the running system. This approach was experimented during the implementation phase and is discussed in the next section. To conclude on the prototype implementation aspects of the mobility data cloud: All above described components are running in our research data center and are available on the Internet.

V. EVALUATION RESULTS

This section describes the trials that were conducted based on the prototype implementation of the mobility data cloud. First the setting for the trials is described, after which the numerical results are presented and discussed, which were obtained based on the conducted experiments.

A. *Experimentation Setup*

The mobility data cloud implementation was deployed on a server (Intel, i686, Dual Core, 2266.747 MHz, 2GB RAM) in our data centre. This server was reachable over the Internet and correspondingly the mobility data cloud could be accessed from a mobile device with an Internet connection. Hence, the goal was set to evaluate the performance of the prototype as perceived from a mobile device, such that an experience can be obtained regarding the feasibility of applying the proposed concepts. In order to gain this experience, an application for automated testing was implemented in Java on an Android 2.3.3 smartphone with 512 MB RAM and a 600 MHz CPU. Thereby, different series of services from the *Service Layer* (**Figure 1** and **Figure 2**) of the mobility data cloud were accessed (from within different concurrent threads) over their belonging REST interfaces. In particular, one thread was implemented that was issuing an "update position" request to the *Position of Electric Vehicles* service every five seconds. A similar rate of position updates could be expected by a vehicle that drives through a city. Simultaneously, a second thread was executed that was randomly picking a service from those on **Figure 2** and was retrieving or pushing data to it over its REST interface.

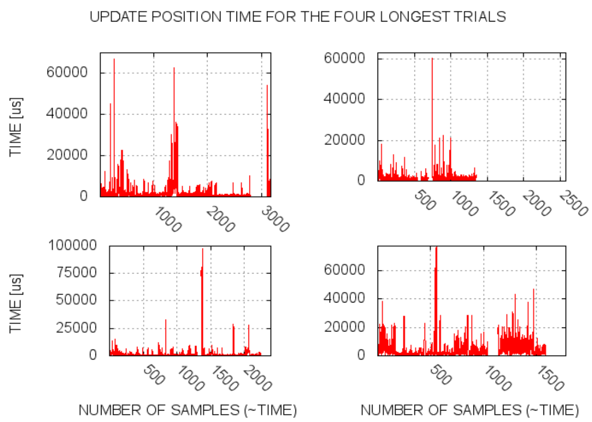

FIGURE 6. MEASURING THE TIME FOR UPDATING THE POSITION OF A VEHICLE

The experiments were conducted by starting the app and driving through the city of Berlin or travelling with the public transport such as busses and trams. The trials started after a first functional version of the services for electric mobility was available. Parallel to the trials, the development team was continuously improving the services, acting as a set of trusted community developers using the opening of the platform at a PaaS level.

B. Numerical Results

The experiments were performed over a period of four months and resulted in 34 drives through the city of Berlin with an overall number of 55511 single measurements, out of which 20697 relate to the position update, and the rest concerning the other services in the Service Layer. Thereby, first the reachability of the services was tested, and secondly the time was measured within which the response of the mobility data cloud service arrived at the mobile device, i.e. at the Android smartphone. That way the interplay of various technological segments was tested within these field experiments. First, it was possible to gain an impression of the reachability of the cloud services over the UMTS/HSDPA/GSM(GPRS) network of the corresponding provider and the Internet backbone behind, whilst driving on the streets of Berlin. In addition, the stability of the implementations and various aspects of the opening of the platform on PaaS level were also evaluated.

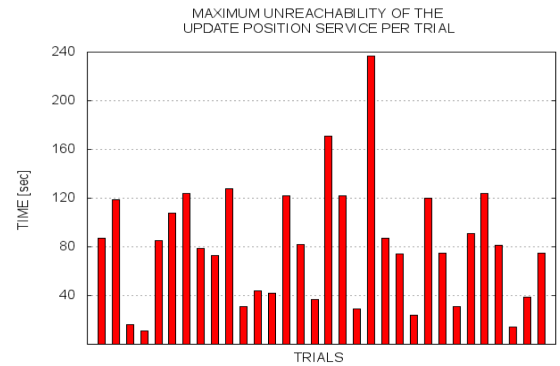

FIGURE 7. MEASURING THE UNREACHABILITY OF THE "UPDATE POSITION" CLOUD SERVICE

Figure 6 shows the time in microseconds that was needed to update the position of the mobile device in the course of the four longest trials (in terms of driving time). One can observe that the time for updating the position was close to 0.1 sec in the worst case. The points where there is no measurement value stand for the temporal unreachability of the cloud service at that moment. In general, the updating of a position was successful in 85% percent of the cases. The unreachability of the *Position of Electric Vehicles* service (and any of the other services) could be explained by the following issues: 1) insufficient mobile network coverage, 2) admission control mechanisms of the mobile network and the network segments behind, and 3) instability of our prototype implementation – as in the upper right trial on **Figure 6**. Given that the updating of a vehicle position is a vital aspect for managing fleets of vehicles, the question should be answered of what is the maximum unreachability period for the *Position of Electric Vehicles* service within the performed experiments. The answer to this question is given in **Figure 7** by a histogram showing the maximum unreachability of the service in seconds, for 32 of the 34 drives. One can see that the maximum unreachability can reach from several (less than 10 seconds) up to almost four minutes. In addition, there were two outliers, where the values were unrealistically high and are considered as bug in an early version of the testing application. However, it can still be claimed that the values between less than 10 seconds and four minutes are acceptable given the complexity of a city road infrastructure, where a vehicle drives through tunnels and under bridges that can hamper the reachability of the mobility data cloud over the 3G network. In addition, it must be mentioned that the services were often crashing due to the opening of the platform on the PaaS level allowing various trusted developers (colleagues from our team) to update independently their services, which has often led to a case where the standalone updated service functioned correctly, but caused problems for the service(s) depending on it. Hence, the lesson can be drawn that the aspect of integration testing should be seriously approached and a new/updated service should be only published after the whole platform in its integrated form has been successfully tested by the automated tests. In general it can be concluded that the opening of the platform on PaaS level requires a sophisticated Quality Assurance process that can ensure the stability of the overall set of mobility services, which are provided by various community developers.

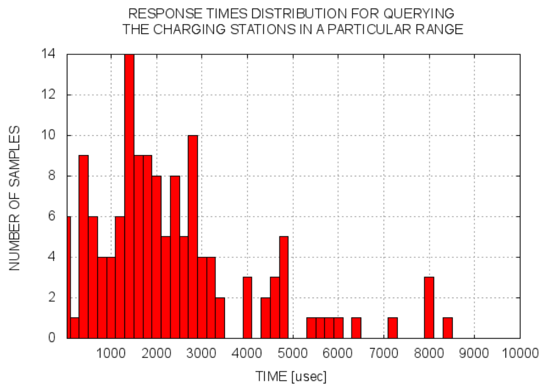

FIGURE 8. RESPONSE TIMES FOR QUERING THE CHARGING STATIONS IN A RANGE

Focusing on additional services, beyond the *Position of Electric Vehicles* service, Figure 8 and **Figure 9** show the distribution of response times for the *Position of Charging Stations* service and the *User Profile* service. Thereby, Figure 8 shows the time which was required for querying the charging stations in a particular range from the Android mobile device – i.e. the *Position of Charging Stations* performance in that case. One can observe that the measured values (in microseconds) are quite acceptable in general, with a very few outliers going up to 0.08 seconds. Moreover, **Figure 9** shows the response times for updating the details for a registered user over the *User Profile* service of the mobility data cloud. The measured values (in microseconds) are again minimal with some very few outliers going up to 0.05 seconds. This observation confirms the claim that the mobility data cloud is reasonably accessible from a mobile device, over a 3G network and the Internet backbone networks behind.

VI. Related Work

With respect to related work, the topic of mobile cloud computing has gained importance within the research community in the past year [1][2][3]. The challenge in that context is to extend the capabilities of end user mobile devices by employing the computational resources of (distant) cloud services, in order to provide computational and data intensive mobile applications. Thereby among others, topics such as Virtual Machine migration, REST services, Quality of Service, and Load Balancing constitute the focus of research in that area. In that line of thoughts, the mobility data cloud proposed here fits into this research scope by providing means for outsourcing, into the cloud, various computations related to the (transport) mobility of people in smart cities. Thereby, the mobility data cloud provides services and tools for aggregating and combining data from different mobility resources and data providers, in order to enable the efficient utilization by end users, accessing the mobility data cloud over smartphones. In addition, the impact of the proposed and evaluated mobility data cloud on ISP networks was elaborated in a previous work [23], as to cover aspects such as QoS and Load Balancing between cloud services.

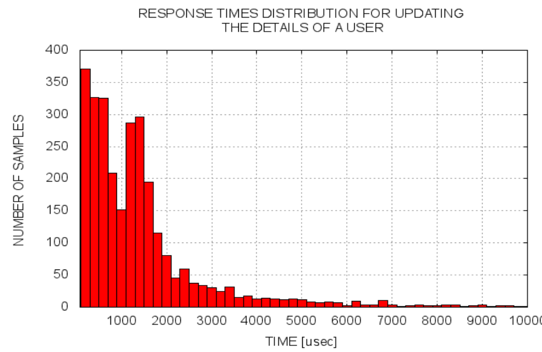

FIGURE 9. RESPONSE TIMES FOR UPDATING THE DETAILS FOR A USER

Furthermore, the topic of V2X (Vehicle-2-X) communication was extensively researched and prepared for deployment [4]. V2X includes various aspects such as the communication between vehicles [5] with the goal of enabling different applications and services, e.g. traffic jam detection. In addition, the communication between a vehicle and (road aside) infrastructure was investigated [6][7] and various applications - such as collision pre-warning and "Green Light Optimal Speed Advisory" - specified and tested [8]. This technology would be very efficient when it is inbuilt and shipped in vehicles in the future. It can be seen as complimentary technology pushing its data to the mobility data cloud, in order to support the provisioning of advanced intermodal resource sharing mobility services in smart cities.

VII. Conclusions

The current paper presented the concept of a mobility data cloud, which is proposed as a solution for consolidating the various fragmented data related to mobility resources, such as vehicles, road infrastructure, maps, electricity charging stations, gas stations etc. Thereby, the mobility data cloud constitutes a cloud based platform, which is accessible over the Internet, and can be reached from mobile devices over the 3G/4G networks and the Internet backbone networks behind. Having described the abstract concept, an architectural design for a mobility data cloud was presented and the dynamic aspects of the architecture were elaborated on a use case from the domain of electric mobility. In addition, a prototype implementation was presented. This prototype was used to test the reachability and performance of the mobility data cloud in the city of Berlin. The results show acceptable reachability (over the mobile network and Internet backbone behind) and response time of the key services. Thereby, besides the infrastructural aspects, the idea of opening the platform on the PaaS level, i.e. allowing trusted developers from the community to implement and improve services, was evaluated within the coding team. Indeed, it was observed that especially the aspect of integration testing should be strongly emphasized when opening such a platform at the PaaS layer, and giving access to developers that do not work in a consolidated team.

With respect to future research directions: The application of the concepts to further case studies, beyond the electric mobility, should be pursued. This could be given for instance by managing bicycle fleets over the cloud, or by applying the prototype for the purpose of mobile surveillance systems such as SPY (Surveillance imProved System) [27].

REFERENCES

[1] Abolfazli, Saeid; Sanaei, Zohreh; Ahmed, Ejaz; Gani, Abdullah; Buyya, Rajkumar (1 July 2013). "Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges". IEEE Communications Surveys & Tutorials 99, pp. 1–32

[2] Fangming Liu, Peng Shu, Hai Jin, Linjie Ding, Jie Yu, Di Niu, Bo Li, "Gearing Resource-Poor Mobile Devices with Powerful Clouds: Architecture, Challenges and Applications";, IEEE Wireless Communications Magazine, Special Issue on Mobile Cloud Computing, vol. 20, no. 3, pp.14-22, June, 2013.

[3] Fernando, Niroshinie; Seng W. Loke, Wenny Rahayu (2013). "Mobile cloud computing: A survey". Future Generation Computer Systems 29: 84–106.

[4] Weiß, Christian. “V2X communication in Europe – From research projects towards standardization and field testing of vehicle communication technology.” Computer Networks 2011 : 3103-3119. Print.

[5] Willke, T.L., P. Tientrakool, and N.F. Maxemchuk, “A survey of inter-vehicle communication protocols and their applications”, IEEE Communications Surveys & Tutorials, 2009

[6] Schweiger, B., Raubitschek, C., Bäker, B. & Schlichter, J, ”ElisaTM – Car to infrastructure communication in the field”, Computer Networks (2011).

[7] Ernst, Thierry, and Arnaud De La Fortelle, “Car-to-car and Car-to-infrastructure Communication System based on NEMO and MANET in IPv6”, in Intelligent Transportation Systems World Congress, 2006

[8] Björn Schünemann, Kay Massow, and Ilja Radusch, "Realistic simulation of vehicular communication and vehicle-2-X applications", In Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems amd workshops (Simutools '08), ICST, Brussels, Belgium, Belgium, Article 62 , 9 pages.

[9] Open Data Platform of the Open Cities project: https://github.com/fraunhoferfokus/opendata-platform, as of date 27.01.2014

[10] Evanela Lapi, Nikolay Tcholtchev, Louay Bassbouss, Florian Marienfeld, Ina Schieferdecker, "Identification and Utilization of Components for a Linked Open Data Platform," 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops, pp. 112-115, 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops, 2012

[11] GovData.de: https://www.govdata.de/, as of date 27.01.2014

[12] Indrawan-Santiago, M., “Database Research: Are We at a Crossroad? Reflection on NoSQL”, 2012 15th International Conference on Network-Based Information Systems, p.45-51.

[13] GeMo Project: http://www.gemo.fraunhofer.de/, as of date 27.02.2014

[14] Jens Klessmann, Philipp Denker, Ina Schieferdecker, Sönke E. Schulz, "Open Government Data Germany:Short Version of the Study on Open Government in Germany", published by Federal Ministry of the Interior in Germany, online available: http://www.bmi.bund.de/SharedDocs/Downloads/DE/Themen/OED_Verwaltung/ModerneVerwaltung/opengovernment_kurzfassung_en.pdf , as of date 27.01.2014

[15] Sarang, Poornachandra. “Practical Liferay: Java-based portal applications development”, book published by Apress, 2009, ISBN-13 9781430218487

[16] JSR 168: Portlet Specification

[17] Jersey 2.5.1 User Guide, online available at: https://jersey.java.net/documentation/latest/user-guide.html, as of date 27.01.2014

[18] Comprehensive Knowledge Archive Network (CKAN): http://ckan.org/, as of date 27.01.2014

[19] Open Knowledge Foundation: http://okfn.org/, as of date 27.01.2014

[20] Apache Tomcat: http://tomcat.apache.org/, as of date 27.01.2014

[21] PostgreSQL: http://www.postgresql.org/, as of date 27.01.2014

[22] Hibernate Community Documentation, Chapter 14. HQL: The Hibernate Query Language, http://docs.jboss.org/hibernate/orm/3.3/reference/en/html/queryhql.htm, as of date 27.01.2014

[23] Nikolay Tcholtchev, Lena Farid, Florian Marienfeld, Ina Schieferdecker, Benjamin Dittwald and Evanela Lapi, "On the Interplay of Open Data, Cloud Services and Network Providers Towards Electric Mobility in Smart Cities", in the proceedings of the International Workshop on GlObal Trends in Smart Cities 2012 October 22-25, 2012, Clearwater, Florida, USA co-located with IEEE LCN 2012

[24] Orri Erling and Ivan Mikhailov, "RDF Support in the Virtuoso DBMS", Networked Knowledge - Networked Media, volume 221 of Studies in Computational Intelligence, pages 7–24. Springer Berlin/Heidelberg, 2009.

[25] Git: http://git-scm.com/, as of date 27.01.2014

[26] Hibernate: http://hibernate.org/, as of date 27.01.2014

[27] SPY architecture: http://www.itea2-spy.org/pdf/SPY2011_posters.pdf , as of date 27.01.2014

[28] Spring framework: http://spring.io/, as of date 27.01.2014

[29] Jenkins Continuous Integration Server: http://jenkins-ci.org/, as of date 12.11.2012